E-Learning Website

Project Report submitted by

V. Sai Varun (420246)

V. Nataraj Gowd (420250)

Github Link: https://github.com/SaiVarun7009/eLearning.git

Department of Computer Science and Engineering,
National Institute of Technology, Andhra Pradesh

06-05-2023

Individual Contributions Breakdown

Introduction: Our project was a web-based platform that provides online courses for students and teachers. The project was completed by a team of two members, with one member responsible for the frontend and backend of user authentication and IT news page, and the other responsible for the frontend and backend related to courses, pathways pages, and enrolling, as well as tracking course progress, course streaming pages, curriculum and references pages, website Responsiveness.

Member 1 (V. Sai Varun - 420246):

- Designed and Developed the frontend and backend architecture for courses, pathways, pages, and enrolling functionality.
- Designed and Developed the frontend and backend architecture for tracking course progress, course streaming pages, curriculum, and references.
- Debugged the frontend and backend of courses, pathways, pages, and enrolling, as well as tracking course progress, course streaming pages, curriculum, and references.
- Implemented website responsiveness, ensuring that the site is easily accessible and navigable on multiple devices.
- Created user-friendly course navigation and streamlining features.
- Tested and debugged all the functionalities of the website, including courses and pathway pages.

Member 2 (V. Nataraj - 420250):

- Designed and Developed the frontend and backend architecture for user authentication, including login, registration, and user roles for students, teachers, and admin.
- Designed and Developed the frontend and backend architecture for the IT news page.
- Ensured the security and privacy of user data by implementing appropriate security measures.
- Provided technical support and troubleshooting for all technical issues related to user authentication and registration.

Conclusion: In conclusion, our project was successfully completed with the collaboration of both team members. Each member made significant contributions in their respective areas, resulting in a fully functional and responsive web platform for online courses. We believe that our project is a testament to the power of teamwork and collaboration in achieving common goals.

Table of Contents

Customer Problem Statement

1.1 Problem Statement

The customer is facing the challenge of providing a streamlined and efficient platform for students to learn computer science subjects. With the abundance of online resources available, the customer recognizes the need for a platform that aggregates the best and most relevant content, reducing the time and effort required for students to find high-quality learning materials.

Specifically, the customer is looking for a solution that provides easy access to the best YouTube videos, GitHub repositories, blogs, and other online resources, curated and organized in a user-friendly manner. Additionally, the customer needs a solution that offers a clear and structured curriculum for computer science subjects, enabling students to understand the path to learning and easily find the best resources for each topic.

Furthermore, the customer requires a platform that allows students to track their progress and easily access the courses they have already completed. The customer also wants a solution that enables teachers to upload and manage their own courses, providing additional learning resources for students.

In summary, the customer is seeking a solution that addresses the challenge of efficiently and effectively learning computer science subjects, reducing the time and effort required to find high-quality learning materials and offering a structured curriculum that enables students to easily track their progress and access relevant resources.

1.2 Decomposition into Sub-Problems

In order to design and develop the eLearning website, we decomposed the overall problem into several sub-problems that can be addressed independently. These sub-problems include:

1. **User Authentication**: This involves creating a system for user authentication and authorization, including login, logout, and registration pages for students, teachers, and admins. This sub-problem also involves designing and implementing password reset and recovery mechanisms.

2. **Course Management**: This involves creating a system for course management, including course creation, editing, deletion, and archiving. It also involves designing and implementing systems for enrolling and dropping courses, tracking course progress, and streaming course content.

3. **Pathway Management**: This sub-problem involves designing and implementing a system for managing pathways, including creating new pathways, adding courses to pathways, and enabling users to browse and search for courses based on the pathways they belong to.

4. **Reference Management**: This sub-problem involves creating a system for managing references, including adding new references, editing and deleting existing references, and enabling users to search for references based on keywords and tags.

5. **User Interface Design**: This sub-problem involves designing and implementing an intuitive and user-friendly interface for the website, including designing and implementing navigation menus, search bars, and interactive elements such as buttons and forms.

Requirements and Analysis

2.1 Functional Requirements

1. **User Authentication**: The system should provide a secure and reliable user authentication mechanism that allows students, teachers, and administrators to access the platform securely.

2. **Registration and Profile Management**: The system should allow users to register and create their profile. Users should be able to update their profile information, such as name, email address, and password.

3. **Course Creation and Management**: Teachers should be able to create, manage, and update courses, including adding and editing course content. Teachers should be able to define course requirements, including prerequisites and learning objectives.

4. **Course Enrollment**: Students should be able to easily enroll in courses they are interested in. The system should allow students to browse courses, view course details, and select courses they want to enroll in.

5. **Curriculum Management**: The system should offer a clear and structured curriculum for computer science subjects, enabling students to understand the path to learning and easily find the best resources for each topic.

6. **Progress Tracking**: The system should enable students to track their progress and easily access the courses they have already completed. Progress tracking should include metrics like completed courses, grades, and achievements.

7. **Resource Aggregation**: The system should provide easy access to the best YouTube videos, GitHub repositories, blogs, and other online resources, curated and organized in a user-friendly manner. The system should enable students to rate and comment on resources.

8. **Notifications**: The system should notify users about new courses, course updates, important deadlines, and achievements.

9. **Course Streaming**: The system should enable course streaming, allowing students to watch videos and other multimedia course materials directly on the platform.

10. **Course Completion Certification**: The system should provide a certificate of completion for students who complete a course.

11. **Course Search**: The system should allow users to search for courses based on keywords, topics, and other criteria.

12. **Pathway Creation and Management**: The system should allow administrators to create

and manage pathways. Administrators should be able to add or remove courses from a pathway, set pathway requirements, and define learning objectives.

13. **Pathway Enrolment**: Students should be able to enrol in a pathway, indicating their interest in a specific subject area. The system should enable students to browse pathways, view pathway details, and select pathways they want to enrol in.

14. **Pathway Progress Tracking**: The system should enable students to track their progress in a pathway. The system should provide students with a clear understanding of what courses they need to take and in what order to complete the pathway.

15. **Pathway Completion Certification**: The system should provide a certificate of completion for students who complete a pathway. The certificate should indicate the pathway name and the courses completed.

16. **Pathway Search**: The system should allow users to search for pathways based on keywords, topics, and other criteria.

17. **Recommended Pathways**: The system should recommend pathways to students based on their interests, previous course history, and other factors.

18. **Administrator Dashboard**: The system should provide an administrator dashboard that allows administrators to manage users, courses, and other system settings

2.2 Nonfunctional Requirements

1. **Usability**: The system should be easy to use and navigate, with an intuitive interface and clear instructions. The system should be designed to minimize the time required for users to learn how to use it.

2. **Performance**: The system should be fast and responsive, with minimal latency and downtime. The system should be designed to handle high volumes of traffic and user activity.

3. **Security**: The system should be secure, protecting user data and preventing unauthorized access to the platform. The system should use encryption, secure authentication mechanisms, and other security measures to protect user information.

4. **Compatibility**: The system should be compatible with a range of devices and browsers, including desktop and mobile devices, and different operating systems.

5. **Scalability**: The system should be designed to handle growth and accommodate additional users and courses as the platform grows.

6. **Reliability**: The system should be reliable, with a high uptime and minimal downtime. The system should be designed to recover quickly from failures and errors.

7. **Data Integrity**: The system should ensure the integrity of data, ensuring that user data is accurate, complete, and consistent across the platform. The system should also provide data backups and disaster recovery mechanisms to protect against data loss.

8. **Interoperability**: The system should be able to integrate with other systems, such as learning management systems or student information systems, to provide a seamless experience for users.

9. **Performance Monitoring**: The system should provide performance monitoring and analytics tools to track system performance, identify bottlenecks and optimize system performance

Use Cases

3.1 Stake Holders

.

1. **Students**: The primary users of the e-learning website, who will use it to enroll in courses, track their progress, and access learning materials.

2. **Instructors**: The users who will create courses, upload course materials, and interact with students through the website.

3. **Administrators**: The users who will manage the website, handle user accounts, and resolve any issues or technical problems.

4. **System Developers**: The team responsible for building and maintaining the e-learning website.

3.2 Actors and Goals

1. **Student**: The initiating actor who will use the e-learning website to enroll in courses, access learning materials, and track their progress towards completion.

2. **Instructor**: The initiating actor who will create and upload courses, add chapters and subchapters to courses, and interact with students through the website.

3. **Administrator**: The initiating actor who will manage user accounts, resolve technical issues, and ensure the smooth operation of the website.

3.3 Use Cases

1. **Register**: The user creates an account on the website, providing their personal information and choosing a username and password.

2. **Login**: The user enters their username and password to log in to their account on the website.

3. **View Courses**: The user browses the list of available courses on the website and selects a course to view. This use case responds to the functional requirement for displaying available courses.

4. **Enroll in Course**: The user selects a course they are interested in and enrolls in the course by providing payment information, if applicable.

5. **Track Progress**: The user can view their progress within a course, including completed

chapters and subchapters.

6. **Create Course**: The instructor creates a new course on the website by providing a course name, description, outcomes, curriculum and other relevant information.

7. **Add Chapter/Subchapter**: The instructor adds a new chapter or subchapter to an existing course, providing a title and description for the new section.

8. **Upload Course Materials**: The instructor uploads course materials such as videos, slides, and assignments to their courses, making them available for students to access.

9. **Administer User Accounts**: The administrator manages user accounts on the website, such as approving new registrations and deactivating accounts as needed.

10. **View Pathways**: The user browses the list of available courses on the website and selects a course to view. This use case responds to the functional requirement for displaying available courses.

11. **Enroll in Pathway**: The user selects a course they are interested in and enrolls in the course by providing payment information, if applicable.

12. **Track Pathway Progress**: The user can view their progress within a course, including completed chapters and subchapters.

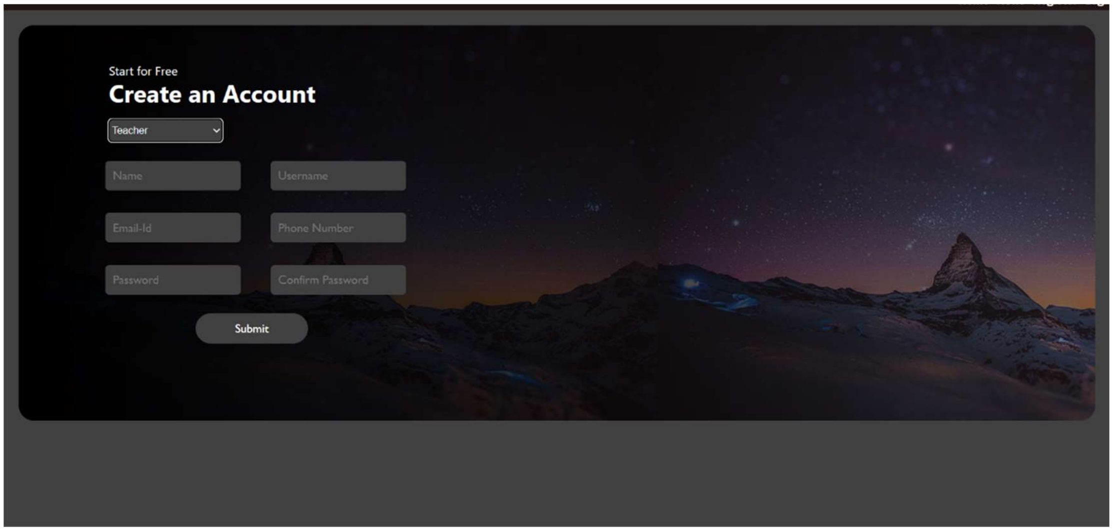
13. **View References**: The user can click on extra resources provided like youtube links, github links, blogs, etc.

User Interface Specification

4.1 Preliminary Design

1. **Register**

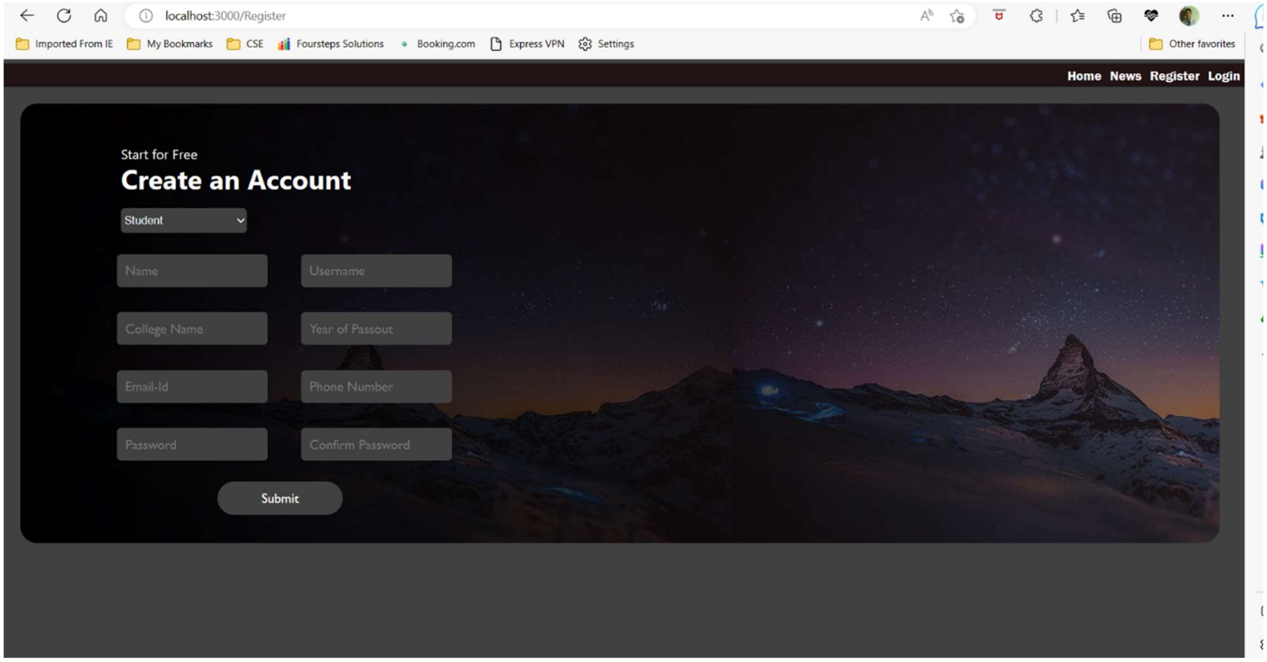
- The user clicks on the "Register" button on the homepage.
- The registration form appears with fields for the user's name, email address, password, and other necessary information.
- The user fills out the registration form and clicks on the "Submit" button.
- The system verifies the information provided by the user and creates a new user account.
- The user is redirected to the homepage and a confirmation message is displayed.
- Track Progress
- The user clicks on the "My Courses" button on the navigation bar.
- The list of courses that the user has enrolled in is displayed.
- The user clicks on the course that they want to track progress for.
- The course overview page is displayed with a progress bar showing the percentage of completion.
- The user clicks on the progress bar to see more detailed information about their progress.
- The system displays a breakdown of completed and remaining chapters and sub-chapters for the course.
-



Start for Free

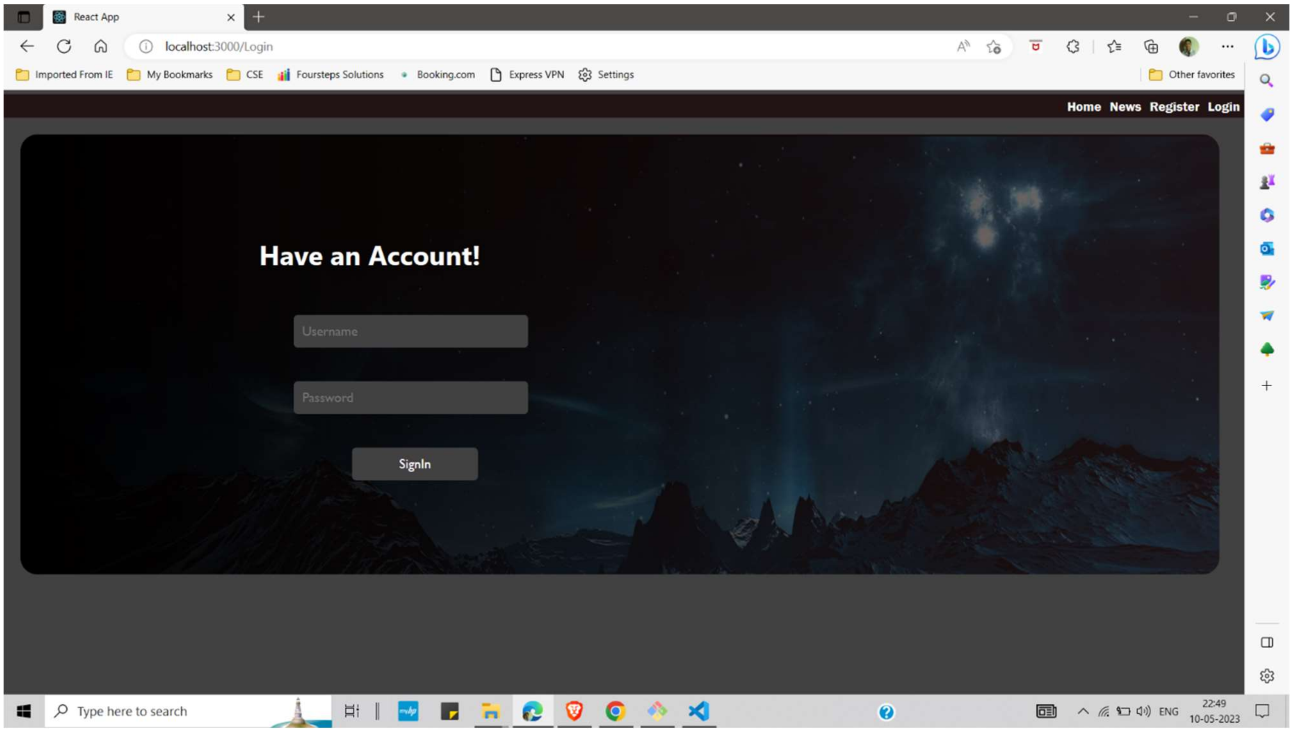
Create an Account

Teacher



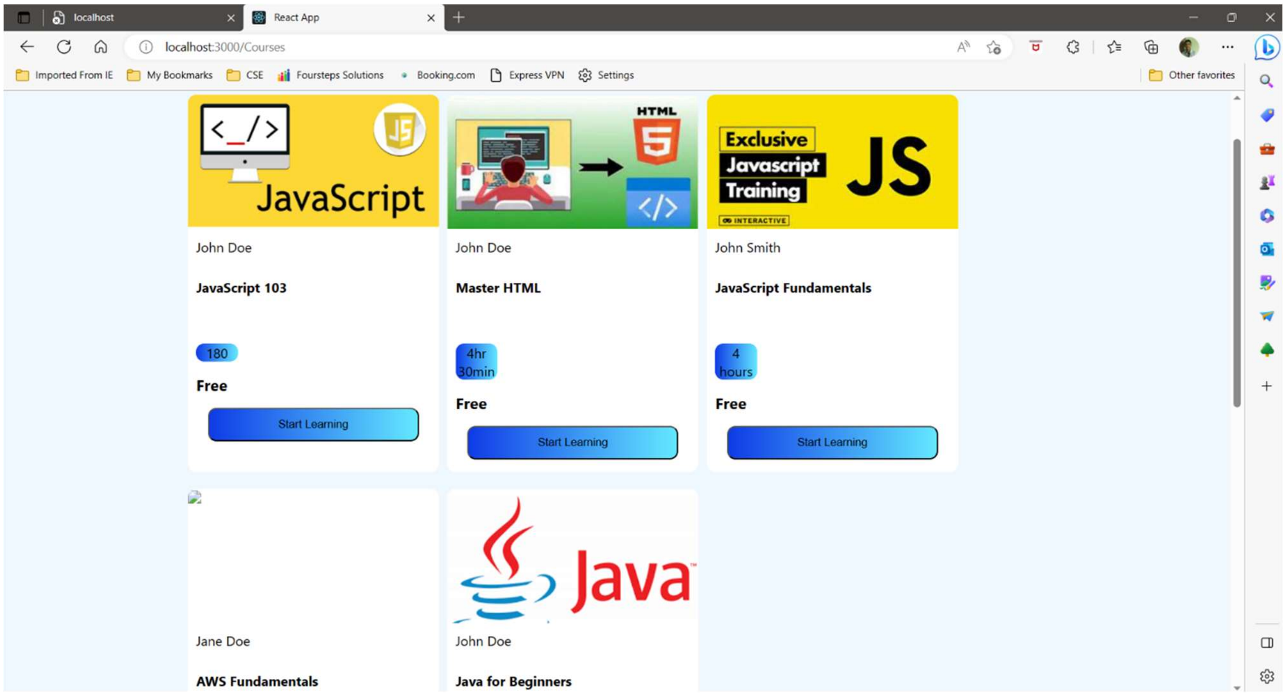
2. Login

- The user visits the login page and enters their credentials (username and password) in the appropriate fields.
- The system verifies the credentials and, if they are correct, redirects the user to the student dashboard.
- If the credentials are incorrect, the system displays an error message and prompts the user to try again.



3. Enroll in Course

- The user navigates to the course catalog page and selects a course they are interested in.
- The system displays the course details, including the course description, curriculum, and references (if any).
- The user clicks the "Enroll" button and the system prompts them to confirm their enrollment.
- Once the user confirms, the system adds the course to their dashboard and displays a success message.



John Doe

JavaScript 103

180

Free

Start Learning

John Doe

Master HTML

4hr 30min

Free

Start Learning

John Smith

JavaScript Fundamentals

4 hours

Free

Start Learning

Jane Doe

AWS Fundamentals

John Doe

Java for Beginners

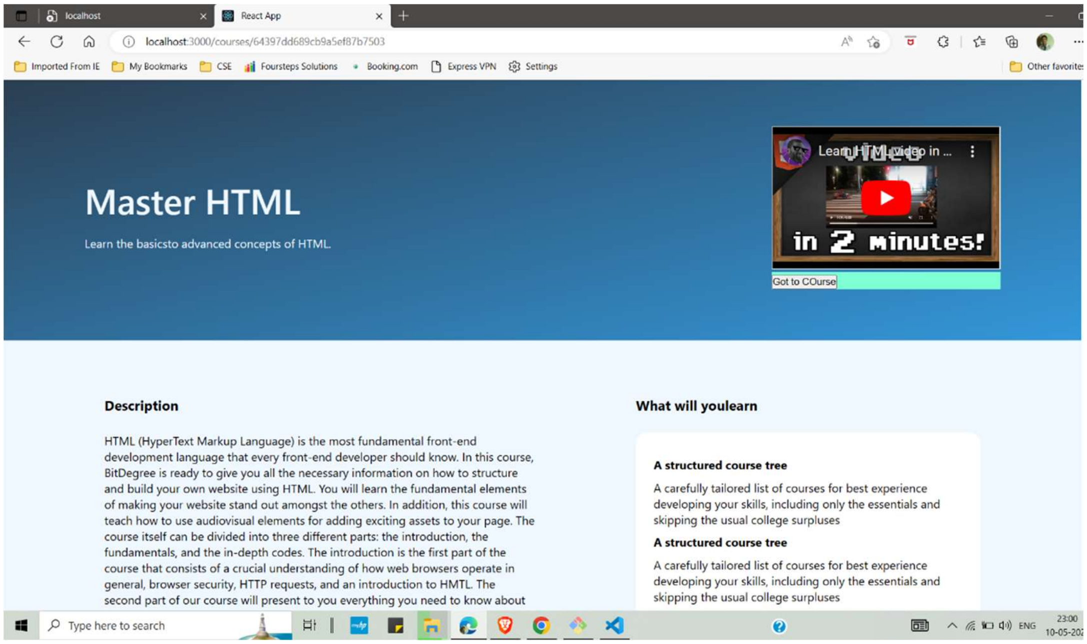
4. Add Chapter to Course

- The teacher navigates to the course editor page and selects the course they want to edit.
- The system displays the course details and prompts the user to select a chapter to edit or add a new chapter.
- The user selects the "Add Chapter" button and enters the chapter details, including the chapter name, description, and references.
- The system adds the chapter to the course and displays a success message.

5. View Courses

- The user clicks on the "All Courses" button on the navigation bar.
- The list of available courses is displayed with brief descriptions and thumbnail images.
- The user clicks on the course they want to view.
- The course overview page is displayed with information about the course, including the instructor, duration, level, and prerequisites.

- The user can see the list of chapters and sub-chapters in the course and click on any of them to view the related content.



Master HTML

Learn the basicsto advanced concepts of HTML.

Description

HTML (HyperText Markup Language) is the most fundamental front-end development language that every front-end developer should know. In this course, BitDegree is ready to give you all the necessary information on how to structure and build your own website using HTML. You will learn the fundamental elements of making your website stand out amongst the others. In addition, this course will teach how to use audiovisual elements for adding exciting assets to your page. The course itself can be divided into three different parts: the introduction, the fundamentals, and the in-depth codes. The introduction is the first part of the course that consists of a crucial understanding of how web browsers operate in general, browser security, HTTP requests, and an introduction to HMTL. The second part of our course will present to you everything you need to know about

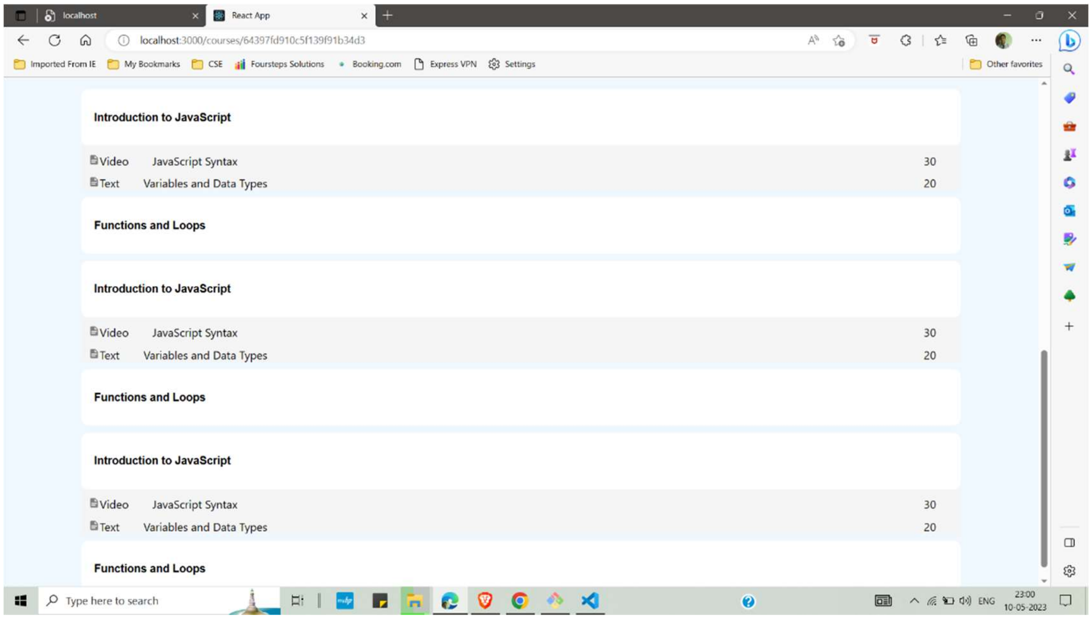
What will youlearn

A structured course tree

A carefully tailored list of courses for best experience developing your skills, including only the essentials and skipping the usual college surpluses

A structured course tree

A carefully tailored list of courses for best experience developing your skills, including only the essentials and skipping the usual college surpluses



Introduction to JavaScript

Video JavaScript Syntax 30

Text Variables and Data Types 20

Functions and Loops

Introduction to JavaScript

Video JavaScript Syntax 30

Text Variables and Data Types 20

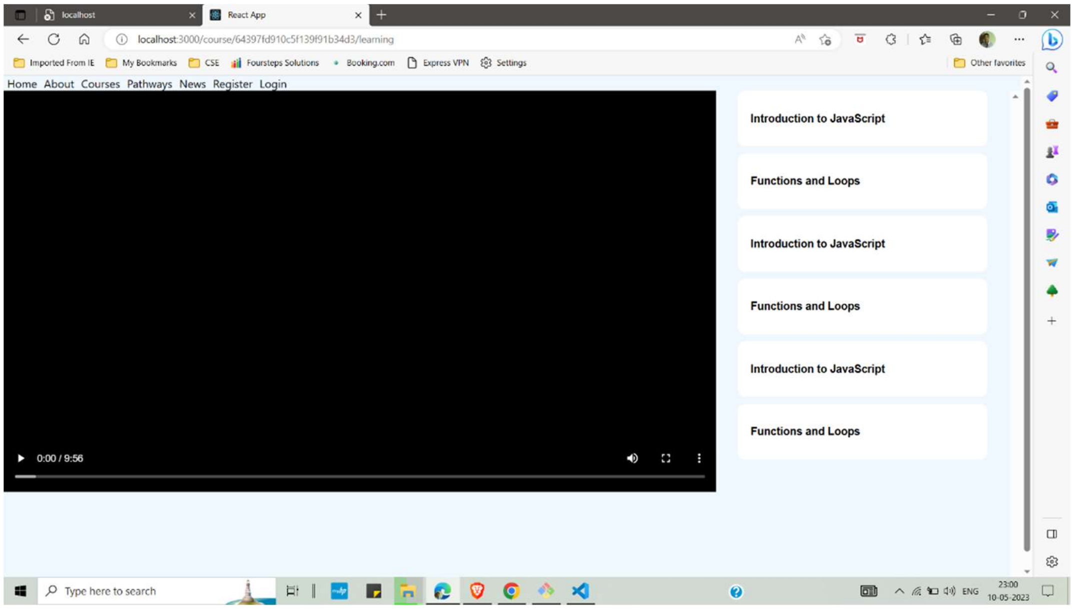
Functions and Loops

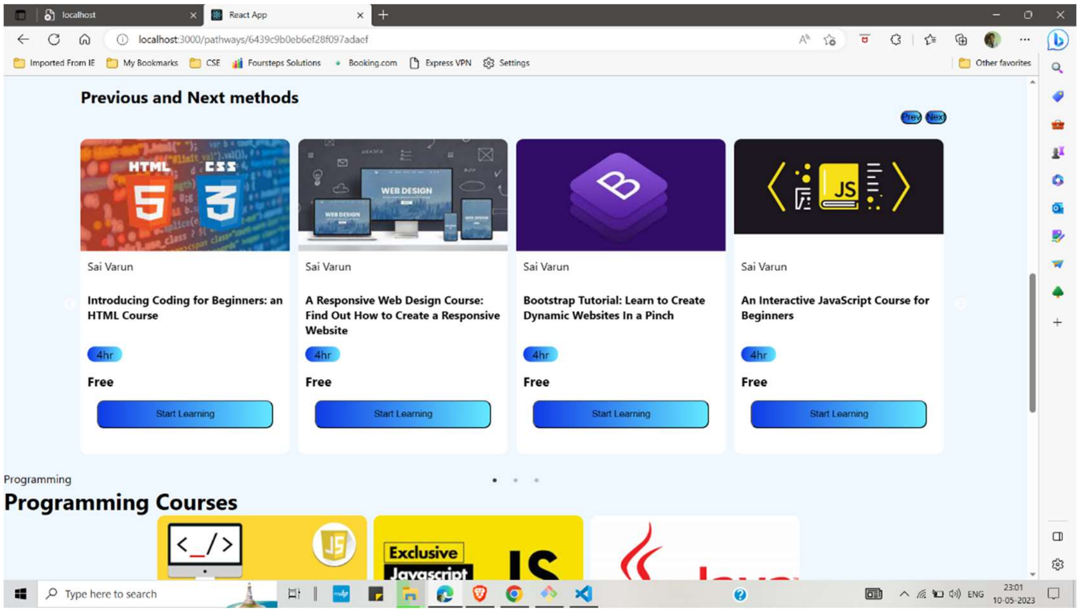
Introduction to JavaScript

Video JavaScript Syntax 30

Text Variables and Data Types 20

Functions and Loops





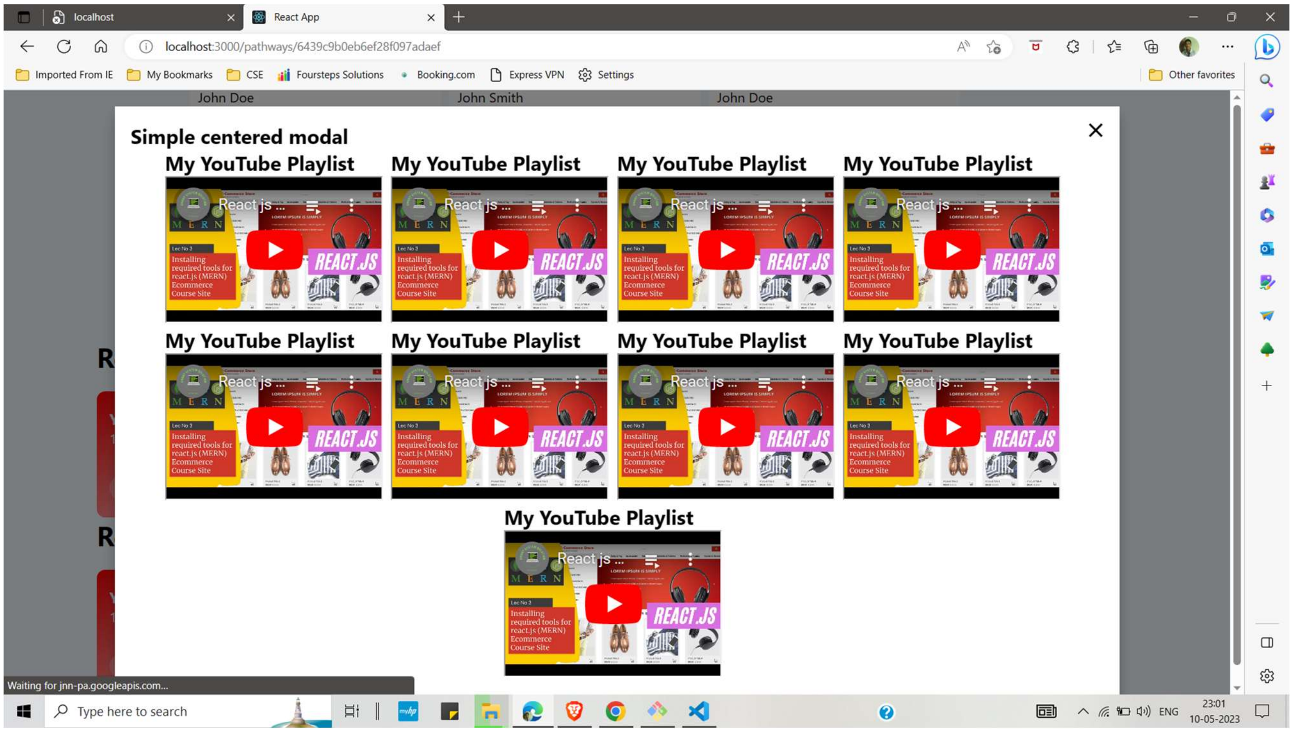
6. Search Courses

- The user types in the keywords related to the course they want to search in the search bar on the homepage.
- The system displays a list of courses related to the search query.
- The user can filter the search results based on the duration, level, and topic of the course.
- The user clicks on the course they want to enrol in.

7. Use Reference Links

- The user clicks on the "References" button on the navigation bar.
- The list of reference links for different topics is displayed.
- The user clicks on the topic they are interested in.

- The list of reference links related to the topic is displayed.
- The user clicks on the link they want to view and it opens in a new tab.



Simple centered modal

My YouTube Playlist

My YouTube Playlist

My YouTube Playlist

My YouTube Playlist

My YouTube Playlist

My YouTube Playlist

My YouTube Playlist

My YouTube Playlist

My YouTube Playlist